



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## **ODHAD ROZESTUPU AUTOMOBILŮ Z MOBILNÍHO TELEFONU UMÍSTĚNÉHO VE VOZIDLE**

VEHICLE FOLLOWING DISTANCE ESTIMATION FROM MOBILE PHONE IN VEHICLE

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**ONDŘEJ ZEMÁNEK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JAKUB SOCHOR**

**BRNO 2017**

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Zemánek Ondřej**

Obor: Informační technologie

Téma: **Odhad rozestupu automobilů z mobilního telefonu umístěného ve vozidle**

**Vehicle Following Distance Estimation from Mobile Phone in Vehicle**

Kategorie: Zpracování obrazu

**Pokyny:**

1. Prostudujte základy zpracování obrazu. Zaměřte se zejména na problematiku detekce objektů a určení vzdálenosti od kamery k objektu známé velikosti.
2. Vyberte vhodnou metodu a navrhnete možnosti řešení dané problematiky.
3. Implementujte mobilní aplikaci umožňující natáčet videa zaznamenávající okolní dění a určovat vzdálenost k nejbližšímu automobilu.
4. Vyhodnoďte přesnost odhadu na vhodně zvolené datové sadě.
5. Vytvořte video prezentující vaši bakalářskou práci.

**Literatura:**

- dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

1. Body 1 a 2, významné rozpracování bodu 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).


Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Sochor Jakub, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



---

doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Cílem této bakalářské práce je mobilní aplikace pro systém Android, která odhaduje vzdálenost vozidel z obrazu kamery mobilního telefonu. Odhad rozestupu je vypočítáván na základě znalostí parametrů kamery, fyzické velikosti vozidla a oblasti snímku, reprezentující detekované vozidlo. Vozidla a jejich registrační značky jsou v obraze detekovány pomocí kaskádových klasifikátorů. Registrační značky jsou detekovány pouze v rámci oblasti detekovaného vozila. Pro detekci vozidel pomocí kaskádového klasifikátoru byla v rámci této práce vytvořena trénovací datová sada. Práce je doplněna o funkci sledování změn rozestupu v čase, která při náhlé změně rozestupu upozorní řidiče zvukovým signálem. Samotný text je pak rozdělen do pěti hlavních částí — porovnání existujících řešení odhadu vzdálenosti, Studium a výběr metod pro rozpoznávání objektů v obraze, návrh aplikace, implementace a vyhodnocení obsahující popis vytváření trénovací datové sady, samotné trénování klasifikátoru a vyhodnocení.

## Abstract

The aim of this bachelor thesis is to create a mobile application for Android that estimates the distance of vehicles based on vehicle size in the camera image of the mobile phone. The estimation of the following distance is evaluated based on known camera parameters, the average vehicle width and the size of image area that represents detected car. Vehicles and their licences plates are detected in the image using cascade classifiers. Licence plate is detected only in the area of the detected vehicle. A training dataset for cascade classifier was created as part of this work. The cascade classifier is designed for vehicle detection. This work is extended with feature that tracks following distance in time and warns you with an acoustic signal on sudden distances change. This thesis is divided into five main parts — comparison of existing solutions for distance estimation, review of object detection methods, application design, implementation and evaluation of detectors, distance evaluation.

## Klíčová slova

odhad vzdálenosti z obrazu, kaskádový klasifikátor, detekce vozidel z obrazu, kalibrace kamery, mobilní aplikace

## Keywords

estimation of distance from image, cascade classifier, vehicle detection from image, camera calibration, mobile application

## Citace

ZEMÁNEK, Ondřej. *Odhad rozestupu automobilů z mobilního telefonu umístěného ve vozidle*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Sochor Jakub.

# **Odhad rozestupu automobilů z mobilního telefonu umístěného ve vozidle**

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jakuba Sochora. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Ondřej Zemánek

16. května 2017

## **Poděkování**

Rád bych poděkoval vedoucímu Ing. Jakubovi Sochorovi za vedení a odbornou pomoc, kterou mi poskytoval v průběhu řešení práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Existující řešení</b>	<b>3</b>
2.1	Měření rozestupu metodou RADAR . . . . .	3
2.2	Light Detection And Ranging (LIDAR) . . . . .	4
2.3	Ultrazvukové senzory pro měření vzdálenosti . . . . .	4
2.4	Odhad vzdálenosti objektu z obrazu . . . . .	5
<b>3</b>	<b>Rozpoznávání objektů v obraze</b>	<b>7</b>
3.1	Kaskádový klasifikátor . . . . .	7
3.2	Local Binary Patterns (LBP) . . . . .	9
3.3	AdaBoost . . . . .	10
<b>4</b>	<b>Návrh aplikace</b>	<b>12</b>
4.1	Identifikace vozidel v obraze . . . . .	12
4.2	Kalibrace kamery . . . . .	13
4.3	Odhad vzdálenosti vozidel v obraze . . . . .	14
4.4	Sledování rozestupu vozidel . . . . .	15
<b>5</b>	<b>Implementace</b>	<b>17</b>
5.1	Mobilní platforma . . . . .	17
5.2	Použité nástroje . . . . .	17
5.3	Uživatelské rozhraní aplikace . . . . .	18
5.4	Záznam videa z kamery mobilního telefonu . . . . .	19
<b>6</b>	<b>Vyhodnocení</b>	<b>21</b>
6.1	Vytvoření trénovací datové sady . . . . .	21
6.2	Trénování kaskádového klasifikátoru . . . . .	23
6.3	Vyhodnocení detekce automobilů . . . . .	24
6.4	Vyhodnocení odhadu vzdálenosti . . . . .	28
6.5	Zhodnocení výsledků . . . . .	32
<b>7</b>	<b>Závěr</b>	<b>33</b>
	<b>Literatura</b>	<b>34</b>

# Kapitola 1

## Úvod

V dnešní době najdeme v automobilech nespočet aktivních či pasivních jízdních asistentů, které přispívají k bezpečnější jízdě. Jednou kategorií těchto asistentů je adaptivní tempomat (Adaptive Cruise Control), který dokáže udržovat rozestup mezi vozidly. Nejčastěji se jedná o systém kombinující radarového měření vzdálenosti, měření pomocí infra kamery a ultrazvukových senzorů. Celý tento systém dokáže měřit vzdálenost s minimální chybou a dokáže měřit vzdálenost mezi vozidly v reálném čase. Bohužel jeho pořizovací cena je dnes stále velmi vysoká a navíc musí být veškeré komponenty integrovány do vozidla.

Tato práce navrhuje alternativní řešení postavené na odhadu rozestupu z obrazu kamery mobilního telefonu. Protože většina chytrých mobilních telefonů dnes disponuje dostatečně kvalitním fotoaparátem pro snímání prostoru před automobilem a výpočetním výkonem, dovolujícím použití algoritmů pro detekci objektů v obraze, je možné využít tuto možnost a vytvořit aplikaci pro mobilní telefon, která bude odhadovat rozestup vozidel. Výsledná aplikace je mnohonásobně dostupnějším řešením pro širší skupinu lidí a zároveň by toto řešení přineslo nezávislost na vozidle, tedy nevyžadovalo by zásah do vozidla.

Hlavním cílem této bakalářské práce bylo vytvoření mobilní aplikace umožňující detekovat rozestup vozidel z kamery mobilního telefonu, vybrat a navrhnout řešení detekce vozidel v obraze a následně vyhodnotit detekci vozidel a odhad vzdálenosti od kamery k vozidlu známé velikosti.

Navržené řešení práce bude postupně rozebráno v jednotlivých kapitolách. V úvodní části se zabývám porovnáním nejpoužívanějších metod pro měření a odhad vzdálenosti. Další kapitola obsahuje návrh samotných komponent aplikace a výběr vhodného detektoru pro detekci vozidel v obraze na mobilním telefonu. V následujících kapitolách se nachází popis implementace jednotlivých částí aplikace pro mobilní operační systém Android, která vás seznámí s použitými nástroji a také uživatelským rozhraním aplikace. V poslední kapitole je popisován postup vytváření datové sady, pro trénování klasifikátoru, podrobné vyhodnocení detektorů vozidel a registračních značek a následně též vyhodnocení výpočtu odhadu rozestupu. V poslední řadě je vedena diskuze nad výsledky detektoru a odhadu vzdálenosti a též nad možnými vylepšeními aplikace.



## Kapitola 2

# Existující řešení

K tomu abychom mohli provést návrh řešení výpočtu odhadu rozestupu vozidel z kamery mobilního telefonu, je nejdříve nutné pochopit a prozkoumat existující metody pro měření rozestupu mezi vozidly. V této kapitole nás budou konkrétně zajímat nejpoužívanější metody měření a odhadu vzdálenosti používané v automobilovém průmyslu, jelikož na tento segment cílí tato práce.

Měření vzdálenosti se v dnešní době provádí různými metodami. Nejčastěji se využívá kombinace více metod pro robustnější a spolehlivější měření v reálném čase. Existující řešení tohoto problému jsou však nejčastěji postavené na základě výpočtu doby šíření vlnového impulsu prostorem odraženého od snímaného objektu. Tyto metody přinášejí většinou vysokou přesnost a spolehlivost měření. Na druhou stranu jsou ale taková zařízení, postavená na tomto principu, většinou velice drahá oproti řešení odhadu vzdálenosti z obrazu. Při použití této metody nám postačí pouze mobilní telefon s fotoaparátem. Avšak i výše popsané metody nacházejí využití v automobilovém průmyslu a proto je vhodné si v následujících podkapitolách nastínit principy, abychom zjistili jejich výhody a nevýhody v porovnání s metodou odhadu vzdálenosti z obrazu, která je použita v této práci a bude podrobně popsána ve třetí kapitole 3.

### 2.1 Měření rozestupu metodou RADAR

První metodou jež se v dnešní době hodně využívá pro měření vzdálenosti je **RADAR** (Radio Detection and Ranging). Jedná se o metodu, která vypočítává vzdálenost na základě doby, kterou urazí vyslaný elektromagnetický impuls k měřenému objektu a zpět.

Radarové zařízení pro měření vzdálenosti obsahuje vysílač a přijímač signálu. Po vyslání signálu, se zařízení přepne do módu příjmu tzv. *echa*, což je výraz pro zpětně odražený signál od objektu. Po přijetí tohoto echa na přijímači zařízení zpracuje signál a na základě času, který uplynul od vyslání a příjmu, lze vypočítat vzdálenost od objektu. RADAR dokáže detekovat více objektů najednou a také není příliš závislá na tvaru či materiálu objektů, na rozdíl od metody odhadu vzdálenosti z obrazu, která je závislá na detekci jednotlivých objektů.

Radary používají pro svojí činnost frekvenci v rozsahu stovek kHz až po 110 GHz. Díky tomu tato metoda dokáže provádět odhad na velmi vysokých frekvencích a pracuje tedy s minimálním zpožděním, může se tedy hodit i pro použití v systémech aktivních bezpeč-

nostních asistentů či systémech adaptivních tempomatů (*Adaptive Cruise Control*<sup>1</sup>), kde zásadním požadavkem pro tyto systémy je právě práce s minimální odezvou.

Maximální dosah radaru silně závisí na výkonu přijímaného signálu odraženého od objektu, který musí být dostatečně velký pro zpracování a následné vyhodnocení.

## 2.2 Light Detection And Ranging (LIDAR)

Další možnost měření vzdálenosti využívající velice podobný princip šíření vlnových impulsů, jako u předchozí metody RADAR, přináší systém **LIDAR** (Light Detection And Ranging). Toto řešení však nepoužívá elektromagnetické impulsy, jako RADAR, ale k měření využívá nízko-výkonového laserového paprsku.

Zařízení typu LIDAR se skládá z laseru, který je použitý jako vysílač, skeneru s přídatnou optikou, fotodetektoru a navigačního systému (GPS).

Laserový vysílač používá ultrafialové, viditelné nebo infračervené pásmo světelné pásmo. Typicky se však využívá rozsah 600 až 1000 nm.

Skener slouží pro usměrnění přijímaných světelných paprsků odražených od objektů. Přijatý signál je směřován do světlovodu, který obsahuje optický filtr. Díky tomu se na fotodetektor dostane pouze požadované světlo ve správném rozsahu pásma. Polovodičový detektor poté zpracuje signál a předá je vyhodnocovacímu zařízení, které z přijatých signálů dokáže sestavit reprezentaci prostředí před zařízením. Díky tomu je pak zařízení schopné detekovat překážky, které se vyskytují před vozidlem.

Stejně jako metoda **RADAR**, dokáže tato metoda měřit i na vzdálenosti přes 150 metrů, a proto jí lze využít pro odhad rozestupu vozidel a to i ve velmi vysokých rychlostech. To je jeden z dalších důvodů, proč tyto metody velmi často najdeme v řešeních automobilových asistentů.

V porovnání s odhadem vzdálenosti z obrazu, je tato metoda přesnější i rychlejší, avšak jedná se o velmi složitý a drahý systém, který tedy není tak dostupný.

## 2.3 Ultrazvukové senzory pro měření vzdálenosti

Rozestup mezi vozidly lze získat i pomocí ultrazvukových senzorů. Ty můžeme najít dnes ve většině moderních automobilů, které disponují alespoň parkovacím asistentem či zmíněným systémem *Adaptive Cruise Control*, který s pomocí těchto senzorů dokáže například udržovat rozestup v kolonách.

Jak už název napovídá, princip této metody těží z odrazu ultrazvukových vln od objektů přes senzor. Použití tohoto zařízení se používá pro měření kratších vzdáleností, kde měřitelná vzdálenost je závislá na výkonu přijímaného signálu.

Nevýhodou tohoto systému je, že některé materiály, které akusticky pohlcují ultrazvuk a tedy neodráží tak dobře signál, jsou špatně detekovatelné touto metodou a znesnadňují měření vzdálenosti.

Tato metoda tedy dokáže měřit kratší vzdálenosti a je velice využívána pro měření rozestupu vozidel v kolonách, kde jsou vozidla maximálně jednotky metrů od sebe, tudíž oproti odhadu rozestupu z obrazu nedokáže měřit na střední a větší vzdálenosti.

---

<sup>1</sup>[http://home.zcu.cz/~rcermak/opvk\\_cm/M\\_05.pdf](http://home.zcu.cz/~rcermak/opvk_cm/M_05.pdf)



## 2.4 Odhad vzdálenosti objektu z obrazu

Předchozí metody fungovaly vždy na podobném principu. Zařízení vyšle vlnový impuls, čeká na jeho návrat a podle času, který byl potřeba k uražení cesty k cizímu objektu a zpět, spočítá výslednou vzdálenost.

V případě odhadu vzdálenosti z obrazu je ale situace úplně jiná. Snímač v kameře, či fotoaparátu nevysílá žádný signál, ale pouze zachytává scénu před snímajícím zařízením. Jedná se tedy o pasivní metodu pro odhad vzdálenosti od objektu a proto nelze použít stejný princip výpočtu vzdálenosti, jako u dosud popsaných řešeních.

Chceme-li tedy změřit vzdálenost k objektu z jediného snímku, musíme vědět rozměr požadovaného objektu, ke kterému tuto vzdálenost měříme. Také se neobejdeme bez detekce samotného objektu v obraze. Jednou z možností, pomocí které dokážeme rozpoznat objekty v obraze, je použití kaskádového klasifikátoru, popsaného v následující kapitole o rozpoznávání v obraze 3. Další možností je využití konvoluční neuronové sítě, avšak tato možnost vyžaduje vyšší výkon.

Pokud tedy dokážeme detekovat automobil v obraze, můžeme, na základě jeho fyzických rozměrů, rozměrů vozidla na snímku a parametrů kamery, odhadnout jeho vzdálenost z obrazu.

Nejznámějším a velice nadějným projektem využívajícím princip odhadu vzdálenosti z obrazu je MobilEye<sup>2</sup>. Tato aplikace skrze analýzu obrazu dokáže předvídat hrozící srážku či měřit bezpečný rozestup mezi vozidly.

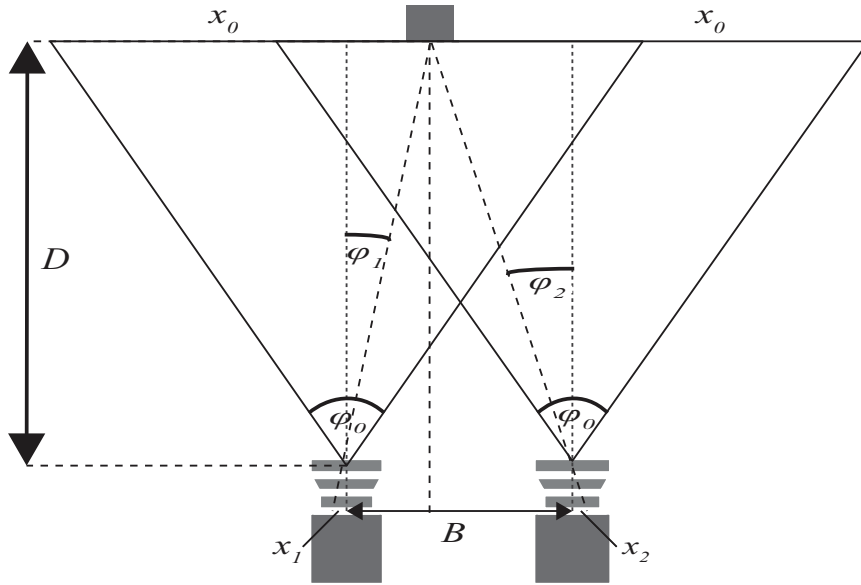
### 2.4.1 Měření vzdálenosti ze stereoskopického obrazu

Pokročilou metodou pro odhad vzdálenosti z obrazu je použití stereoskopického kamerového systému. Tedy využití páru kamer, které snímají stejnou scénu pod odlišným úhlem (*Toe-in*) a nebo rovnoběžně (*Off-axis*). Tato metoda je podrobně popsána v článku [9]. Snímání prostoru dvěma kamerami, nám umožňuje vytvořit prostorový vjem trojrozměrných předmětů, jejich vzdálenost a hloubku v prostoru. Pomocí těchto metod lze ze snímků obou kamer, vypočítat fyzickou velikost výsledného objektu. Z toho důvodu je tento princip, na rozdíl od využití jedné kamery, výhodnější pro obecnější použití.

Při použití metody *Toe-in*, znázorněné na obrázku 2.1, jsou pohledy obou kamer soustředěny do stejného bodu. Scéna je tedy zachycena pod různými úhly, proto je výsledný obraz pro člověka nepřirozený.

---

<sup>2</sup><http://www.consumerreports.org/cro/magazine/2013/11/safety-upgrades-for-your-car/index.htm>



Obrázek 2.1: Schéma stereoskopického systému s použitím páru rovnoběžně nastavených kamer (*Off-axis*)

Princip *Off-axis* spočívá v rovnoběžném nastavení kamer a jejich odsazením o vzdálenost  $B$ . Protože kamery jsou namířené stejným směrem, záběry jednotlivých kamer se kryjí a obraz působí přirozeně. Proto se tato metoda využívá nejčastěji.

$$D = \frac{B \cdot x_0}{2 \cdot \left(\frac{\varphi_0}{2}\right) \cdot (x_2 - x_1)} \quad (2.1)$$

Kde,  $B$  je vzdálenost mezi senzory jednotlivých kamer,  $\varphi_0$  je úhel záběru kamer,  $(x_1 - x_2)$  je rozdíl mezi dvěma oblastmi snímku objektu v pixelech a  $x_0$  je celková šířka snímku jedné kamery v pixelech.

Ze vzorce 2.1 je vidět, že vzdálenost  $D$  je nepřímo úměrná poměru  $(x_1 - x_2)$ , velikosti objektu na jednotlivých snímcích. Díky tomu, jak už bylo zmíněno, nepotřebujeme pro výpočet vzdálenosti fyzický rozměr objektu.

## Kapitola 3

# Rozpoznávání objektů v obraze

K tomu, abychom byli schopni zautomatizovat detekci vozidel v obraze, je zapotřebí najít vhodnou metodu, která je použitelná na mobilních telefonech, které nedisponují moc vysokým výpočetním výkonem.

Metod pro detekci objektů v obraze je mnoho. Některé z nich detekují objekty na základě analýzy barev, s využitím různých barevných modelů [12], nebo analýzy obrazového histogramu. Další možností je detekce na základě tvarů objektů, tímto problémem se zabývá například Houghova transformace popsaná v tomto článku [5]. Neposledním řešením může být také vyhledávání objektů s použitím předdefinované šablony, které se využívá například pro vyhledávání znaků v obraze. Tato metoda byla publikována v práci Petra Ciky a spol. [3].

Všechny tyto zmíněné metody se ale hodí spíše pro detekci jednodušších objektů, které mají velmi podobnou barvu, tvar a velikost. Například pro rozpoznávání dopravního značení, registračních značek vozidel. Nehodí se ale pro detekci obecnějších objektů, jako je automobil, který může mít velmi odlišnou barvu, velikost i tvar.

Pro detekci obecnějších typů objektů se využívají komplexnější a pokročilejší metody. První z nich je kaskádový klasifikátor, který je podrobně popsán v následující podkapitole. Další možnou metodou je sestavení a natrénování konvoluční neuronové sítě<sup>1</sup>. Každopádně i přes možnosti dosažení vyšší přesnosti touto metodou, není vhodné využít tuto metodu pro detekci vozidel v této práci, protože vyžaduje vysoký výkon, kterým mobilní telefony zatím stále nedisponují. Princip konvolučních neuronových sítí byl podrobně popsán v článku [8].

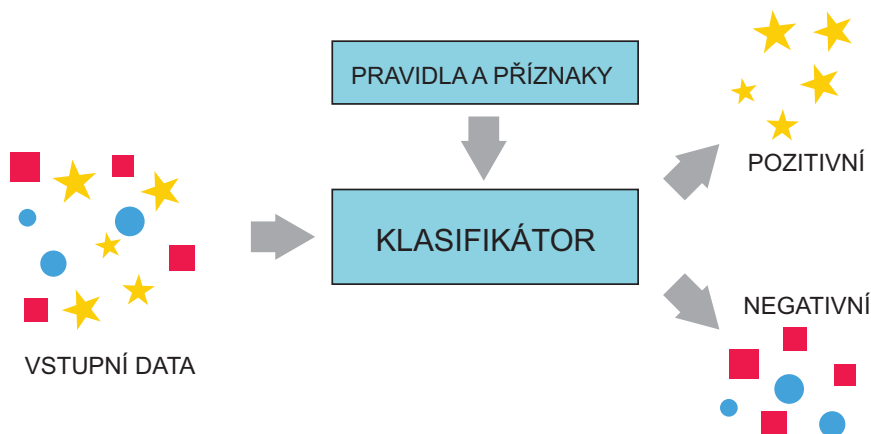
### 3.1 Kaskádový klasifikátor

Úkolem kaskádového klasifikátoru je třídění vstupních dat na pozitivní a negativní detekce na základě příznaků, získaných během jeho trénování. Princip byl podrobněji popsán v článku [10].

Pro lepší představu si můžeme představit klasifikátor jako kontrolního pracovníka u pásu, který má na starosti odstraňovat z linky vadné či vzhledově nevyhovující výrobky. Tento pracovník na základě instrukcí pohledem rozezná, zdali se jedná o bezvadný výrobek či nikoli. Nevyhovující výrobky odstraní z pásu a na výstup linky tedy vyfiltruje pouze výrobky splňující všechny požadavky kvality, pro které byl pracovník proškolen. Viz obrázek 3.1.

---

<sup>1</sup><http://cs231n.github.io/convolutional-networks/>

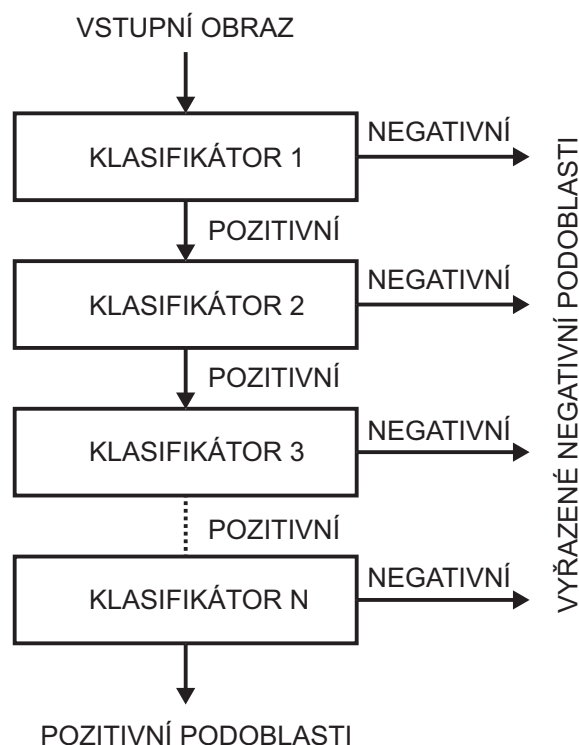


Obrázek 3.1: Příklad zjednodušeného klasifikátoru

Kaskádový klasifikátor můžeme stejně jako popsaného pracovníka proškolit na určitý typ objektů, tomuto procesu se říká **trénování klasifikátoru**. Trénovací data pro klasifikátor by v příkladu pracovníka linky představovaly dvě množiny snímků. V jedné, tzv. **pozitivní množině**, by se nacházely snímky zachycující bezchybný výrobek. Naopak tzv. **negativní množina** by byla tvořena snímky nevyhovujících výrobků a snímky na kterých se nevyskytují bezchybné výrobky a zachycují prostředí ve kterém bude detektor nasazen.

Kaskádový klasifikátor je tvořen kaskádou tzv. slabších klasifikátorů, které klasifikují obrazové objekty do pozitivní nebo negativní třídy. Do pozitivní třídy klasifikují hledané obrazové objekty a do negativní zbylé oblasti. Části obrazu z pozitivní třídy následně předají na vstup dalšího slabého klasifikátoru a tímto způsobem je obraz na vstupu postupně klasifikován, dokud neprojde celou kaskádou. Na výstupu pak získáme oblasti hledaných objektů, neboli pozitivně ohodnocené oblasti obrazu všemi slabými klasifikátory.

Hledaný objekt v tomto případě je nutné chápat jako segment vstupního obrazu, který projde všemi úrovněmi klasifikátoru (Viz obrázek 3.2), je podobný pozitivním vzorkům z trénovací sady klasifikátoru a obsahuje tedy s velkou pravděpodobností hledaný objekt.



Obrázek 3.2: Znázornění kaskádového klasifikátoru

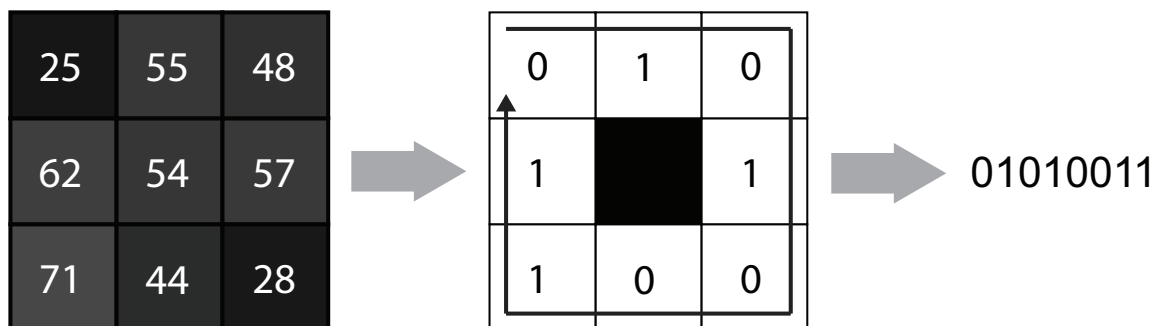
Pro vytvoření kaskádového klasifikátoru je zapotřebí vytvořit datovou sadu pozitivních a negativních vzorků. Pozitivní vzorky v datové sadě reprezentují objekt, který chceme hledat v obraze a pro který tedy chceme vytvořit klasifikátor. Naopak negativní snímky slouží jako pozadí, pro trénování. Nesmí tedy zachytávat objekty, které chceme pomocí klasifikátoru detekovat.

Přesnost tohoto rozpoznávání kaskádovým klasifikátorem závisí především na použité trénovací datové sadě, která obsahuje výše popsanou tzv. pozitivní sadu snímků požadovaného objektu k detekci a negativní sadu obrázků, kterých se požadovaný objekt nevyskytuje.

## 3.2 Local Binary Patterns (LBP)

Local Binary Patterns [1] se používají pro jednoduchou a rychlou analýzu obrazu pomocí diskrétních příznaků, v tomto případě histogramu. Jedná se o strukturní operátor, který převádí vstupní obraz do výstupního, složeného z odstínů šedi na základě struktury vstupního obrazu. LBP vytváří postupně pro každý pixel vektor příznaků skládající se z hodnot okolních pixelů.

Při vytváření jednotlivých příznaků se prochází každý pixel obrazu a získává se jeho hodnota pomocí porovnávání s jeho sousedními pixely  $3 \times 3$ , kdy střední pixel představuje práh. Pokud je hodnota sousedních pixelů větší, jeho příznak bude mít hodnotu 1, pokud je naopak menší, než prahová hodnota, jeho hodnota bude 0. Viz obrázek 3.3.



Obrázek 3.3: Princip získávání příznakového vektoru prostředního pixelu z příznaků okolních pixelů

Po získání výsledku prahování, z nichž je vytvořen vektor příznaků, se vektor vynásobí danými vahami (Viz obrázek 3.4) a tím získáme výslednou hodnotu LBP pro prostřední pixel.

0	1	0
1		1
1	0	0

 $\times$ 

1	2	4
8		16
32	64	128

 $=$ 

0	2	0
8		16
32	0	0

$$\text{LBP} = 2 + 8 + 16 + 32 = 58$$

Obrázek 3.4: Násobení vektoru a získání výsledného příznaku prostředního pixelu

Výsledné binární hodnoty pro každý z okolních pixelů složíme do jednoho binárního čísla. Při skládání bereme první hodnotu z levého horního rohu a ve směru hodinových ručiček přidáváme další hodnoty, jak je zobrazeno na schématu šipkou 3.3. Výsledkem je LPB histogram.

### 3.3 AdaBoost

AdaBoost (*Adaptive Boosting*) je klasifikační algoritmus založený na principu strojového učení boosting. Využívá se pro zlepšení vlastností obecných algoritmů strojového učení. Tato metoda je schopna natrénovat s trénovací množinou dat kaskádový klasifikátor, tím, že zkombinuje několik slabých klasifikátorů. Tomuto algoritmu se věnuje odborný článek [11].

Princip tohoto algoritmu využívá tzv. slabé klasifikátoru (*Weak learner*) s velmi nízkou úspěšností jen o málo větší než u náhodného rozhodování. AdaBoost pracuje na vstupu s trénovací množinou dat s příznaky, zdali se jedná o pozadí nebo hledaný objekt. Slabé klasifikátory jsou na konci trénování spojeny v jeden mnohonásobně silnější klasifikátor (*Strong learner*). Přesnost výsledného detektoru se odvíjí od použité trénovací množiny.



Klasifikační algoritmus pro učení slabých klasifikátorů  $h_t(x)$ , které jsou vybírány z množiny klasifikátorů  $H$ . Následně při jejich lineární kombinaci vzniká jeden silný a nelineární klasifikátor  $H(x)$ . Vstupní trénovací množina klasifikátoru  $S$ , složená z dvojic  $(x_i, y_i)$ , kde  $x_i$  označuje hodnotu příznaku a  $y_i$  udává odpovídající třídu příznaku  $i$ ,  $y_i \in \{-1, 1\}$  pro  $i = 1, \dots, M$ .  $M$  značí velikost trénovací množiny<sup>2</sup>.

AdaBoost používá váhování trénovací množiny pomocí vah  $D_t$ , jenž jsou na začátku trénování nastaveny rovnoměrně. V každé další smyčce algoritmu se vybere takový slabý klasifikátor, který disponuje nejmenší chybou klasifikace s aktuálním nastavení vah  $D_t$ , zároveň ověří, že chyba není větší, než chyba náhodného klasifikátoru, tedy větší než 50 % a následně algoritmus provede výpočet koeficientu slabého klasifikátoru v lineární kombinaci  $H(x)$  a na závěr aktualizuje hodnoty všech vah  $D_t$  trénovací množiny.

Samotný výběr slabého klasifikátoru se řídí zmíněným pravidlem, že jeho chyba  $\varepsilon_t$  nesmí překročit hodnotu 0,5. To by totiž mohlo způsobit, že algoritmus přestane pracovat správně. Vybrán je takový klasifikátor, který bude lépe klasifikovat dosud špatně klasifikovaná měření, kterým je zvětšena váha a naopak dobře klasifikovaným měřením je tato váha snížena. Díky tomu tento algoritmus dokáže snižovat chybu detekce exponenciálně s rostoucím počtem zapojených klasifikátorů.

V extrémních případech, kdy je počet klasifikátorů příliš vysoký, může dojít k tzv. *přetrénování*, stavu, kdy výsledný klasifikátor ztratí schopnost generalizace detekcí a není tedy schopen rozeznávat nic jiného, než konkrétní trénovací pozitivní vzorky.

---

<sup>2</sup><http://rob.schapiro.net/papers/explaining-adaboost.pdf>

## Kapitola 4

# Návrh aplikace

V této kapitole je popsán návrh prostředků k vytvoření mobilní aplikace odhadující rozestup mezi vozidly. Podíváme se na způsob, jak aplikace detekuje vozidla v obraze, jak odhaduje vzdálenost od vozidel před kamerou a z jakých důvodů a jakým způsobem provádí kalibraci kamery mobilního telefonu. Taktéž jsou zde popsány požadavky uživatelského rozhraní a návrh rozšiřující funkce sledování změn rozestupu vozidel v čase. Viz obrázek 4.1.



Obrázek 4.1: Jednotlivé bloky návrhu aplikace

Techniky měření vzdálenosti pracující na principu šíření vlnových impulsů, zmíněné v předchozí kapitole existujících řešení 2, sice dosahují velmi dobrých výsledků, avšak ani jedna se příliš nehodí pro řešení této práce. A to z důvodů, že tato zařízení mají většinou nekompaktní rozměry, bylo by příliš komplikované je nainstalovat tak, aby dokázala komunikovat s mobilním telefonem a v neposlední řadě by tento proces vyžadoval instalaci zařízení do vozidla.

Proto se při návrhu aplikace zaměříme na řešení s využitím metody odhadu vzdálenosti v obraze. To nám umožní využít k odhadu rozestupu vozidel pouze samotný mobilní telefon s kamerou a dostatečným výpočetním výkonem.

### 4.1 Identifikace vozidel v obraze

Základem pro odhad vzdálenosti k objektu v obraze, je identifikování jeho pozice na snímku a především detekce rozměrů oblasti objektu na snímku, kterou samotný objekt zabírá. Popsání principu použitého pro identifikaci objektů popisuje předchozí kapitola 3.

Pro detekci vozidel bylo nutné vybrat vhodnou techniku, která bude splňovat předem definované požadavky. Jedním z důležitých parametrů při výběru detektoru vozidel v obraze, je spolehlivost detekce. Dále je také nutné vybrat výkonnostně nenáročný detektor,

který bude možné použít i na mobilních telefonech s nižším výkonem. Z toho také vyplývá další požadavek rychlost detekce, protože pro sledování změn rozestupu vozidel hraje důležitou roli čas, za který je algoritmus schopný detekovat vozidlo v obraze.

Detektor pro identifikaci vozidel v obraze, který splňuje vytyčené požadavky, je kaskádový klasifikátor (popsaný v kapitole 3.1) natrénovaný datovou sadou, která obsahuje tisíce snímků automobilů a nákladních vozidel. Klasifikátor je velice flexibilní možností pro řešení tohoto problému, jelikož ho lze natrénovat k detekci libovolného typu objektu a jeho rychlost detekce na mobilních telefonech je dostatečná. Spolehlivost detekce klasifikátoru je však velmi závislá na trénovací sadě a proto je nutné věnovat tomuto problému velkou pozornost. Postup vytváření této trénovací datové sady je popsáno v kapitole 6.1.

Taktéž pro detekci **registračních značek** vozidel v obraze byl použit kaskádový klasifikátor. Avšak jelikož byla tato možnost zvolena až v průběhu implementace aplikace, nebyla pro tuto možnost vytvořena trénovací datová sada, ale byl použit již vytvořený klasifikátor z demo příkladů projektu opencv<sup>1</sup>.

## 4.2 Kalibrace kamery

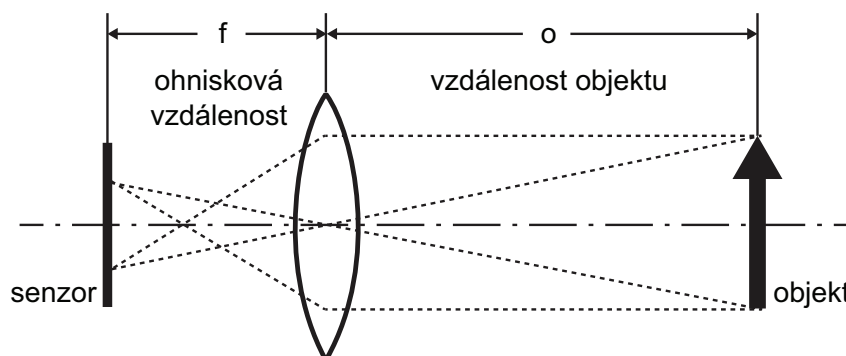
Pokud chceme odhadovat vzdálenost ze snímku kamery, musíme mít informace o parametrech snímáče a objektivu, který byl použit při pořizování snímku. Při použití rozdílných objektivů s jinou ohniskovou vzdáleností, se výsledný obraz promítaný na snímáč liší v úhlu záběru, tudíž zachycené automobily jsou na výsledném snímku v rozdílné velikosti. Pokud například použijeme objektiv s ohniskovou vzdáleností 50 mm, který přibližně odpovídá svým úhlovým záběrem lidskému oku, a změříme rozměr objektu na snímku, dostaneme jiný rozměr, než v případě použití objektivu 18 mm, kde bude výsledná velikost v pixelech menší, protože na obrázku dostaneme větší oblast.

Dalšími důležitými parametry, které je nutné zohlednit, jsou fyzické rozměry snímáče a jeho rozlišení v pixelech (Viz obrázek 4.2). Fyzický rozměr je dalším parametrem, který udává výslednou oblast záběru focenou oblast. Z rozlišení senzoru můžeme zase zjistit, jak velkou oblast zabírá vozidlo v poměru k celému snímku. Což se nám bude hodit při výpočtu rozestupu vozidel popsaném v následující kapitole 4.3.

Při výpočtu vzdálenosti je tedy zapotřebí počítat s rozdílnými parametry kamery mobilních telefonů [6].

---

<sup>1</sup>[https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\\_licence\\_plate\\_rus\\_16stages.xml](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_licence_plate_rus_16stages.xml)



Obrázek 4.2: Princip kalibrace kamery (Zobrazení objektu na senzoru snímáče kamery)

### 4.3 Odhad vzdálenosti vozidel v obraze

Abychom mohli měřit velikost rozestupu vozidel z kamery mobilního telefonu umístěného za čelním sklem automobilu, potřebujeme znát informace o fyzické velikosti vozidla, ke kterému měříme vzdálenost.

Zde nastává ale hlavní problém, jelikož z jediného snímku kamery nemáme možnost zjistit přesné rozměry vozidel, které budeme detekovat v obraze a následně k nim odhadovat vzdálenost, tudíž je zapotřebí vymyslet jiný způsob, jak zjistit jejich velikost.

Situace by byla jiná, kdybychom měli k dispozici stereoskopickou sadu dvou kamer, s jejichž využitím by bylo možné odhadovat vzdálenost ze snímků, aniž bychom předem znali fyzické rozměry vozidla. Principy této techniky jsou popsány v kapitole 2.4. Této problematice se rozsáhle zabýval Bc. Martin Janeček ve své diplomové práci [7].

Bohužel ale drtivá většina mobilních telefonů nedisponuje stereoskopickým viděním a proto je zapotřebí nalézt jiné řešení tohoto problému.

Jelikož na silnici můžeme narazit na spoustu rozdílných typů automobilů, je důležité nalézt jejich nejvíce společný rozměr, který jsme schopni vyčíst ze snímku kamery a jehož hodnota se napříč různými typy vozidel příliš nemění.

Z detekovaných oblastí vozidel v obraze dokážeme zjistit jejich výšku a šířku. Výška automobilu se může velice často lišit, příkladem může být nákladní automobil, který je mnohem vyšší než osobní vozidlo, tím pádem není vhodné zvolit výšku automobilu jako společný jmenovatel pro odhad vzdálenosti.

Využít můžeme ale také šířku vozidla, která se samozřejmě také u různých vozidel liší, ale rozdíl je minimální, na rozdíl od výšky vozidla. Průměrná šířka dnešních vozidel je 1,8 m<sup>2</sup>. Tato hodnota nám tedy ve výpočtu vzdálenosti může zastoupit fyzickou šířku vozidla, viz následující podkapitola 4.3.1.

#### 4.3.1 Výpočet vzdálenosti vozidla v obraze

Samotný výpočet vzdálenosti mezi vozidly je popsán vzorcem 4.1, který vypočítává vzdálenost na základě velikosti tzv. bounding boxu (popsaný v kapitole 5.3.1) ohraničující nalezené vozidlo na snímku, jak je vyobrazeno zeleným obdélníkem na obrázku z aplikace 4.3, znalosti o průměrné šířce automobilu a informacích o parametrech kamery.

<sup>2</sup>[http://usatoday30.usatoday.com/money/autos/2007-07-15-little-big-cars\\_N.htm](http://usatoday30.usatoday.com/money/autos/2007-07-15-little-big-cars_N.htm)

Stejný výpočet se používá i pro výpočet vzdálenosti registrační značky automobilu, s tím rozdílem, že místo šířky vozidla v obraze se používá šířka registrační značky a taktéž fyzická šířka vozidla je nahrazena fyzickou šířkou registrační značky, která je 520 mm.

**Vzorec pro výpočet vzdálenosti z obrazu:**

$$d = \frac{W_1 \cdot F_l}{\frac{W_2}{S_1} \cdot S_2} [mm] \quad (4.1)$$

kde:

$W_1$  ... fyzická šířka automobilu v milimetrech

$F_l$  ... ohnisková vzdálenost v milimetrech

$W_2$  ... šířka vozidla v pixelech

$S_1$  ... šířka senzoru v pixelech

$S_2$  ... šířka senzoru v milimetrech

Jak bylo již zmíněno, fyzická šířka automobilu je tedy v aplikaci zastoupena průměrnou šířkou vozidla. Parametry kamery, jako ohnisková vzdálenost, fyzická šířka senzoru a jeho rozlišení, jsou získány samotnou mobilní aplikací. Poslední proměnná, šířka vozidla v obraze, je získána na základě šířky detekované oblasti (bounding boxu), ohraničující vozidlo, viz obrázek 4.3.



Obrázek 4.3: Ukázka z aplikace - odhad vzdálenosti na základě velikosti automobilu

## 4.4 Sledování rozestupu vozidel

Doplňující součástí aplikace, která dokáže předpovídat nebezpečné situace ze změn rozestupu mezi vozidly je sledování změn rozestupu v čase. Se znalostí těchto změn je možné aplikací rozpoznávat, zdali vozidlo jedoucí před námi náhle začalo brzdit či do našeho jízdního pruhu nevjelo vozidlo. Z toho důvodu je nutné implementovat algoritmus, který by byl

schopný porovnávat rozestup v čase a na základě toho odhadovat, zdali se nejedná o příliš rychlou a tím pádem nebezpečnou změnu rozestupu.

Pro samotné sledování rozestupu vozidel je potřeba znát hraniční rychlost přibližování, která závisí především na aktuální rychlosti vozidla. Tato hraniční rychlost udává jak moc se může rozestup snížit za jednotku času, aniž by to bylo nebezpečné, což může nastat například v situaci při zmíněném náhlém brzdění vozidla před námi. Proto je také důležitou součástí aplikace integrace měření rychlosti automobilu pomocí mobilního telefonu s využitím GPS dat.

Algoritmus sledování rozestupu vozidel tedy při odhadu rozestupu k vozidlu ukládá odhadnutou vzdálenost. Při další detekci sleduje oblast ve které se předchozí detekce objevila a pokud je znovu klasifikováno vozidlo ve stejné oblasti, tak zkontroluje odhadnuté vzdálenosti a čas, který od předchozího odhadu uběhl a porovná tento výsledek k hraniční rychlosti pro aktuální rychlost vozidla. Pokud se tedy při dalším měření vzdálenosti z obrazu nebezpečně sníží rozestup od vozidla, aplikace varuje uživatele krátkým zvukovým signálem.

Hraniční rychlost přiblížení  $v_t$  (Viz vzorec 4.2) je vypočítána z aktuální rychlosti vozidla  $v_a$  a frekvence detekce vozidel  $f_o$  v obraze mobilní aplikací, což je hodnota závislá na aktuálním výkonu mobilního telefonu.

$$v_t = v_a \cdot \left(1 - \frac{1}{f_o}\right) \quad (4.2)$$



## Kapitola 5

# Implementace

Tato část se zabývá implementací mobilní aplikace, jejíž návrh je popsán v předchozí kapitole. Důležitou součástí této sekce je výběr použité mobilní platformy, pro kterou je aplikace implementována. Tato kapitola také popisuje použité nástroje pro zpracování obrazu z kamery, implementaci detektoru vozidel a odhadu vzdálenosti, postup získání aktuální rychlosti z GPS a vytvoření vhodného uživatelského rozhraní. Taktéž je zde popsán záznam videí.

### 5.1 Mobilní platforma

Pro mobilní aplikaci byl zvolen operační systém **Android**, který splňuje požadavky návrhu. Hlavním důvodem pro zvolení této mobilní platformy byla široká podpora a dostupnost programovacích nástrojů a knihoven, které byly využity pro vytvoření aplikace. Tomu také přispěl fakt, že se neustále zvyšuje podíl systému Android na trhu<sup>1</sup>. Jelikož se jedná dnes o nejrozšířenější mobilní platformu, je také možné předpokládat větší popularitu a taktéž větší uživatelskou zpětnou vazbu výsledné aplikace, která je velmi důležitá.

Vytvořená aplikace je implementována primárně v programovacím jazyku **Java**, který je podporovaný systémem Android. Z důvodu rychlejšího testování a experimentování s detektorem, odhadem a sledováním vzdálenosti, byla aplikace implementována také v jazyce **Python**, který svou jednoduchou syntaxí a snadným použitím externích knihoven značně usnadnil celou implementaci, odladění i návrh aplikace.

### 5.2 Použité nástroje

Použité nástroje pro implementaci aplikace, odhadující vzdálenost na základě detekce vozidel z obrazu, jsou převážně z open source knihovny **OpenCV** (Open source Computer Vision). Jedná se o knihovnu s veřejně dostupným kódem, která obsahuje stovky funkcí pro zpracování a analýzu obrazu. Mezi tyto funkce patří také podpora kaskádových klasifikátorů, které lze s pomocí knihovny natrénovat. Stejně tak máme možnost využít funkcí pro následné využití klasifikátoru pro detekci objektů v obraze. Z této knihovny byly také využity funkce pro filtrování obrazu.

---

<sup>1</sup><http://www.businessinsider.com/smartphone-market-share-android-ios-windows-blackberry-2016-8>

Další použitou knihovnou je **ffmpeg**<sup>2</sup>, která umožňuje záznam videa a zvuku. Díky této knihovně bylo možné nahrávat video i s upravenými snímky aplikací. Tedy nejen vytvořit samotný záznam z kamery mobilního telefonu, ale též zahrnout do videa i označené detekované vozidla a jejich vzdálenosti.

### 5.3 Uživatelské rozhraní aplikace

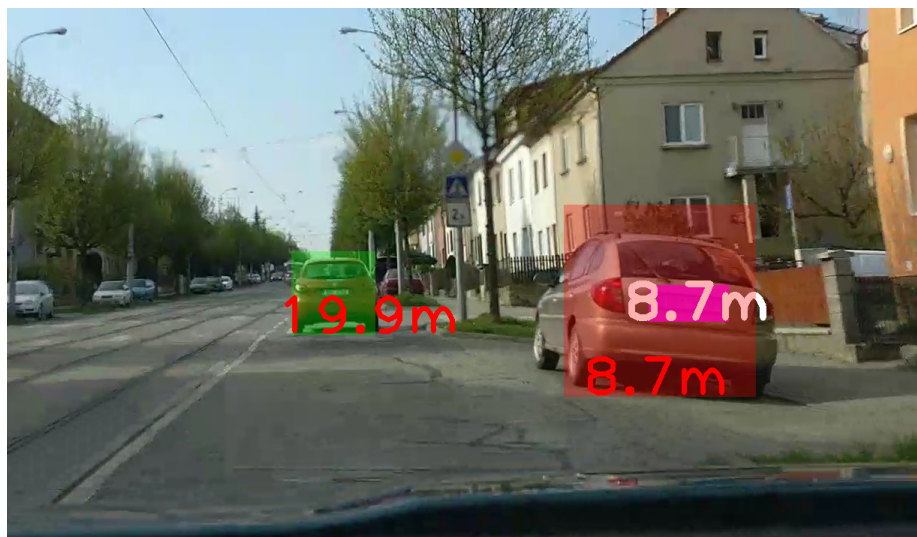
Jelikož se jedná o aplikaci určenou primárně pro řidiče, byl při vytváření uživatelského rozhraní kladen důraz na jednoduché prostředí aplikace, které by bylo snadno pochopitelné, disponovalo by jednoduchým ovládáním a nevyžadovalo by pozornost od uživatelů po spuštění detekce či nahrávání.

Abychom splnili tyto požadavky, bylo nutné co nejvíce nastavení zautomatizovat a uživateli nabídnout již připravenou aplikaci, která po zapnutí tedy nevyžaduje zdržující nastavování parametrů fotoaparátu, pro kalibraci kamery (popsáno v kapitole 4.2), či volby oblasti detekce.

To bylo také důvodem k implementaci získávání parametrů mobilního telefonu z vestavěných funkcí použitého operačního systému Android. S jejich pomocí byl implementováno automatické doplnění parametrů kamery do výpočtu odhadu vzdálenosti popsaného v kapitole 4.3. Stejně tak jsou získávány informace o rozlišení displeje, pro který aplikace přizpůsobí výslednou reprezentaci vozidel a jejich rozestupů.

#### 5.3.1 Reprezentace automobilů a jejich rozestupu

Abychom mohli navrhnout reprezentaci automobilů a jejich rozestupů, je nutné nejdříve popsat situace, které mohou nastat viz následující tabulka 5.1 a obrázek 5.1.



Obrázek 5.1: Ukázka interpretace detekce vozidel a rozestupů

Uživatelské rozhraní zprostředkovává uživateli obraz z kamery mobilního telefonu, ve kterém jsou následně označeny aktuálně detekované automobily a případně jejich detekované registrační značky.

<sup>2</sup><https://www.ffmpeg.org/documentation.html>

Situace	barva ohraničujícího boxu	barva odhadu rozestupu
Detekovaný automobil	zelená	červená
Rozpoznaná registrační značka	růžová	bílá
Nebezpečné snížení rozestupu	červená	černá

Tabulka 5.1: Seznam situací reprezentovaný v uživatelském prostředí

Oblasti rozpoznané jako vozidla, či registrační značky jsou na snímcích reprezentovány pomocí barevných obdélníků, jejichž rozměry udávají výstupní struktury funkce pro detekci kaskádovým klasifikátorem.

Odhad rozestupu detekovaných vozidel je reprezentován jako celé číslo s jednotkou vzdálenosti a nachází se vždy v levé spodní části detekované oblasti.

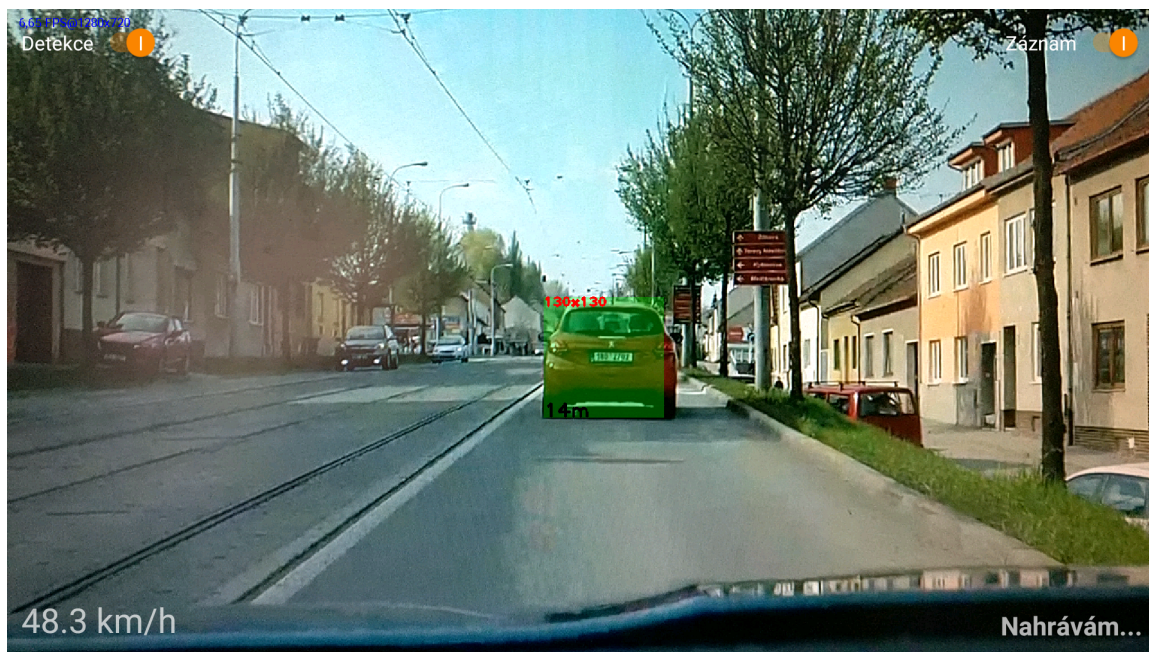
Pro informování o nebezpečném přibližování rozestupu vozidel byla zvolena změna barvy ohraničujícího boxu vozidla. Barvy reprezentující jednotlivé situace naleznete v tabulce 5.1.

### 5.3.2 Signalizace nebezpečné změny rozestupu vozidel

Jak bylo již zmíněno, aplikace je cílená zejména na řidiče vozidel, kteří by se při používání aplikace měli věnovat především řízení. Z toho důvodu není pouhá grafická signalizace nebezpečných situací ideálním řešením a bylo tedy zapotřebí implementovat i jiné upozornění. Proto je použito také zvukové upozornění, které řidiče upozorní při náhlé změně rozestupu mezi vozidly.

## 5.4 Záznam videa z kamery mobilního telefonu

Součástí práce je také možnost záznamu videa z jízdy. Záznam lze spustit a vypnout pomocí tlačítka v pravém horním rohu aplikace. po zahájení nahrávání se v pravém spodním rohu zobrazí informace o nahrávání (Viz obrázek 5.2). Po dalším stisknutí tlačítka záznam se záznam ukončí. Video záznam se ukládá ve sníženém rozlišení  $640 \times 480$ , kvůli optimalizaci a především vyššímu výkonu detektoru. Aplikace při vytváření záznamu ukládá výsledný video soubor do složky `/pilote_app/` v kořenovém adresáři mobilního telefonu. Vytvořený záznam z jízdy se ukládá s příponou `*.mp4` a název souboru je ve generován na základě aktuálního času, ve formátu `YYYYMMDDHHmmss`.



Obrázek 5.2: Ukázka ze spuštěné aplikace

## Kapitola 6

# Vyhodnocení

Vyhodnocení aplikace bylo prováděno ze dvou hlavních pohledů. První se zaměřuje na ohodnocení spolehlivosti a přesnosti detektoru vozidel na snímcích z reálného provozu. Druhá část vyhodnocení je poté věnována odhadu vzdálenosti.

V první podkapitole se ale nejdříve podíváme na popis a vytváření trénovací datové sady pro detekci vozidel, díky které bylo možné natrénovat kaskádový klasifikátor pro detekci vozidel.

K získání výsledků detektoru byly natočeny záznamy z reálného provozu na dálnici, silnicích druhé a třetí třídy a taktéž z městského provozu. A to z důvodu lepšího vyhodnocení detektoru na reálných datech. Výsledky a podrobný postup je popsán v následující kapitole [6.3](#).

Také pro dostatečně kvalitní vyhodnocení odhadu rozestupů byly pořízeny snímky **vozidel z přesně naměřených vzdáleností**. Každé vozidlo, zachycené ve vytvořené referenční sadě, bylo nafoceno v různých vzdálenostech podmínkách a různými mobilními telefony. Podrobnější popis vytváření této sady pro vyhodnocení odhadu vzdálenosti naleznete v podkapitole [6.4](#).

### 6.1 Vytvoření trénovací datové sady

Pro kvalitní výsledky kaskádového klasifikátoru vozidel v obraze bylo potřeba vytvořit trénovací datovou sadu. Vytvořením dostatečně velké a rozmanité datové sady docílíme tím přesnější a robustnější detekce. Kaskádový klasifikátor a princip jeho trénování byl popsán v kapitole [3.1](#).

Datová množina byla pro tuto práci získána převážně z anotovaného video datasetu **TME Motorway Dataset** [\[2\]](#), který obsahuje přes 27 minut videa z kamery, umístěné za čelním sklem osobního vozidla, snímající provoz na dálnici. Při vytváření této video sady byl využit laserový scanner, pro detekci osobních a nákladních vozidel. Informace z detekce byly zaznamenány do anotačních souborů k jednotlivým snímkům videa a bylo tedy možné ze snímků získat mnoho vzorků vozidel, které se použily v datové sadě pro trénování kaskádového klasifikátoru.

Jako pozitivní vzorky, zachycující pouze samotná vozidla, byly použity z velké části snímky automobilů získané z již zmíněných záznamů datasetu MotorwayDataset. Snímky vozidel byly vyříznuty ze snímků videí pomocí malé aplikace vytvořené v rámci této práce. Další množinu vzorků tvořila vozidla získaná z hierarchické databáze obrázků **ImageNet**. O této databázi se můžete více dočíst v odborném článku [\[4\]](#).



Negativní snímky, které neobsahují vozidla, byly získány také z již zmíněné databáze obrázků **ImageNet**. Ta obsahuje obrovské množství hierarchicky rozdělených fotografií a obrázků do kategorií, které udávají objekt či situaci vyobrazenou na obrázcích. Díky tomu bylo možné získat velké množství pozitivních a negativních snímků podle specifikovaných požadavků (např. budov, přírody, apod).

Přesnost detekce klasifikátoru závisí na popisované trénovací datové sadě, která se tedy skládá ze dvou částí – **pozitivní sady** snímků požadovaného objektu k detekci a **negativní množiny** obrázků neobsahující objekty k detekci. Počet negativních snímků by měl být mnohonásobně větší, než pozitivních.

### 6.1.1 Pozitivní vzorky

Pro trénovací datovou sadu byly použity pozitivní vzorky zachycující různé druhy vozidel, viz obrázek 6.1. Sada obsahuje zhruba 200 typů automobilů. Osobní automobily jsou zachyceny na 70 % pozitivních snímcích, zbytek je tvořen nákladními vozidly, karavany a přívěsy.



Obrázek 6.1: Ukázka pozitivní sady vozidel

Při tvorbě pozitivních vzorců byl kladen důraz na to, aby se ve vzorcích nevyskytovalo příliš pozadí nebo pokud možno v co nejmenší míře. Pokud by totiž při vytváření datové sady nebyl brán ohled konkrétnost pozitivních vzorků, docházelo by po natrénování k nepřesné detekci vozidla a nebylo by možné přesně měřit vzdálenost z velikosti detekované oblasti.

Při sbírání pozitivních vzorků se vybíraly snímky zachycující zadní části vozidel. Důvodem pro toto rozhodnutí byl fakt, že při výsledné klasifikaci je důležitá především klasifikace vozidel jedoucích před kamerou, tzn. taková, která jsou zachycena kamerou zezadu. Kdybychom totiž při tvorbě zanesly do trénovací sady i ostatní části vozidel, nastal by problém s rozlišením mezi hrozícím nebezpečím a projíždějícím automobilem v jedoucím opačném směru.

Taktéž bylo důležité vybírat do množiny automobily v různých barvách s různým jasnem. Na samotné barvě nezáleží, jelikož se před trénováním klasifikátoru všechny vzorky převádí do šedých odstínů, každopádně jas dané barvy zůstává i po převodu do odstínů šedé.

### 6.1.2 Negativní vzorky

Negativní snímky byly vybrány na základě prostředí, pro kterou je klasifikátor určen. Datová sada je tvořena fotografiemi přírody (30 %), budov (30 %), ulicemi bez automobilů



(15 %) a silnicemi (15 %). Zbylou část tvoří také snímky oblohy a mraků, jelikož se téměř vždy objevují na klasifikovaných snímcích.

## 6.2 Trénování kaskádového klasifikátoru

Jak již bylo popsáno v kapitole o návrhu aplikace 4, pro detekci vozidel v obraze byl zvolen kaskádový klasifikátor. K jeho vytvoření je potřeba vytvořit datovou sadu. Její vytváření je popsáno v předchozí podkapitole 6.1. Následující tabulka 6.1 obsahuje informace o použitých pozitivních a negativních snímcích, využitých pro natrénování kaskádového klasifikátoru.

Typ datové sady	počet vzorků
Pozitivní snímky	2922
Negativní snímky	8444

Tabulka 6.1: Snímky trénovací sady

Samotné trénování klasifikátoru předchází vytvoření vektorového souboru pozitivních snímků pomocí funkce `opencv_createsamples` z OpenCV knihovny. Tato funkce na základě předložených parametrů (Viz tabulka 6.2) a souboru s anotacemi pozitivních snímků, vybere označené oblasti, změní jejich velikost podle vstupních parametrů a následně všechny takto získané vzorky uloží do požadovaného binárního souboru, který může být následně předložen trénovací funkci.

Použité příkazy pro trénování kaskádového klasifikátoru:

Parametry vytváření vektoru pozitivních snímků	
počet pozitivních vzorků ( <code>-num</code> )	2922
šířka výstupních vzorků ( <code>-w</code> )	50
výška výstupních vzorků ( <code>-h</code> )	50

Tabulka 6.2: Parametry vytváření vektoru pozitivních snímků

Po vytvoření binárního souboru, který obsahuje pozitivní snímky lze přejít k vytváření klasifikátoru. Při spouštění funkce `opencv_traincascade` je zapotřebí zadat parametry pro specifikaci trénování. Z tabulky 6.3 je možné vyčíst, že při trénování je použita funkce *LBP* pro popis obrazu, popsaná v kapitole 3, při jejím použití je trénování a detekce mnohonásobně rychlejší, nežli při použití Haar příznaků.

Použité parametry pro trénování kaskádový klasifikátor	
počet pozitivních vzorků ( <code>-numPos</code> )	2600
počet negativních vzorků ( <code>-numNeg</code> )	8444
počet úrovní klasifikátoru ( <code>-numPos</code> )	20
šířka trénovacích vzorků ( <code>-w</code> )	50
výška trénovacích vzorků ( <code>-h</code> )	50
použitá funkce pro analýzu obrazu ( <code>-featureType</code> )	LBP

Tabulka 6.3: Parametry použité pro trénování kaskádového klasifikátoru

### 6.2.1 Vylepšování klasifikátoru

Trénování klasifikátoru bylo během diplomové práce několikrát opakováno s rozšířenou datovou sadou na základě průběžného vyhodnocování na záznamech z reálného provozu a statických fotografií.

Hlavními důvody k natrénování klasifikátoru na novém datasetu byly především časté falešné detekce objektů, podobných nákladním vozidlům. Například zadní části velkých orientačních tabulí na dálnicích a silnicích první třídy, které se jim velmi podobají. Taktéž nastávaly falešné detekce u dělených oken budov, které se svým tvarem také přibližují nákladním vozidlům.

Při zjištění tohoto problému falešných detekcí byla negativní množina doplněna o dalších 500 snímků cedulí a o více než 900 snímků oken získaných opět z databáze **ImageNet**.

## 6.3 Vyhodnocení detekce automobilů

Pro vyhodnocení detektoru (kaskádového klasifikátoru) pro detekci vozidel byly použity vytvořené záznamy z reálného provozu. Samotné vyhodnocení klasifikace je rozděleno na dvě hlavní části — vyhodnocení **pozitivní klasifikace** a vyhodnocení **negativní klasifikace**.

Při vytváření těchto záznamů z reálného provozu byl brán ohled na různorodost okolí a typu komunikací. Pořízené videa zachycují jízdu v městském provozu, po dálnici a silnicích druhé a třetí třídy.

Při zkoumání pozitivní klasifikace sledujeme části snímku, které klasifikátor detekoval jako vozidla. Pokud tato klasifikovaná část snímku zachycuje automobil, do vyhodnocení zaznamenáme tuto klasifikaci jako **True positive**, což označuje správnou detekci. Pokud ale detekovaná část snímku vozidlo neobsahuje, hodnotíme tuto detekci, jako **False positive**, označující falešnou detekci.

Druhou částí vyhodnocení je hodnocení negativní klasifikace. V této části hodnotíme segmenty snímku, které klasifikátor nedetekoval jako vozidlo. V tomto případě **True negative** označuje oblasti, které neobsahují automobil a které detektor správně nedetekoval. V opačném případě, kdy mezi detekovanými oblastmi snímků chybí ty, která vozidla obsahují, označujeme tento případ termínem **False negative**.

Pro tuto práci nedává smysl v hodnocení negativní klasifikace kontrolovat **True negative** situace, jelikož nemáme možnost zaznamenat na snímcích všechny možné části, které neobsahují vozidlo. Proto vyhodnocení úspěšnosti klasifikátoru probíhá pouze pro výše popsané zbylé tři situace.



Obrázek 6.2: Ukázka z anotovaného datasetu snímků ze záznamů reálného provozu

Pro vyhodnocení bylo použito 300 snímků z natočených videí. Vybrán byl vždy jeden snímek pro každé dvě sekundy, aby jednotlivé snímky nebyly příliš podobné a předešlo se tak zkreslení výsledků. Ukázka z množiny získaných dat je na obrázku 6.2.

Následně byly vytvořeny anotace vozidel a jejich registračních značek pro každý obrázek. V anotacích byly pokaždé zaznamenány všechny vozidla na snímku, s tím rozdílem, že vozidla, které se nevyskytovala přímo před vozidlem, byla navíc označena příznakem *ignore*, který ve vyhodnocování ignoruje detekce v této označené oblasti.

Poté proběhla detekce vozidel, kdy se zaznamenávaly detekované oblasti a jejich skóre klasifikace.

Následně se detekce porovnaly s anotacemi a k jednotlivým detekcím se přiřadily IoU (*Intersection over Union*) hodnoty.

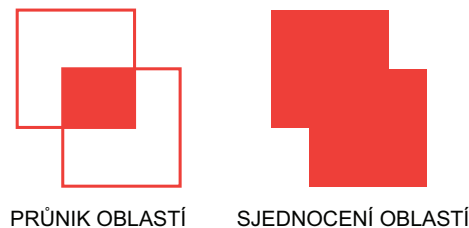
Hodnota **IoU** (*Intersection over Union*) reprezentuje přesnost detekce oblasti objektu v porovnání s anotovanou oblastí objektu (Viz obrázek 6.3). Výpočet **IoU** je uveden v následujícím vzorci 6.1.

$$IoU = \frac{A_o}{A_u} \quad (6.1)$$

Kde:

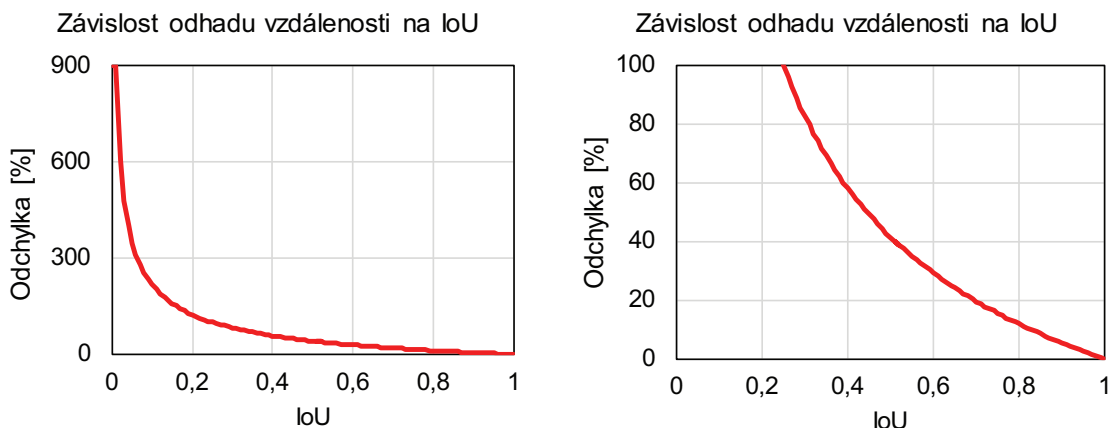
$A_o$  ... oblast průniku anotované a detekované oblasti.

$A_u$  ... oblast sjednocení anotované a detekované oblasti.



Obrázek 6.3: Ilustrace operací nad oblasti  $A_o$  a  $A_u$  k výpočtu **IoU**

Na základě této hodnoty byly ve vyhodnocení, mezi **True positive**, vyselektovány pouze detekce, které disponují hodnotou **IoU** větší, než 0,8. Tento práh byl zvolen, na základě závislosti odhadu vzdálenosti na **IoU**, jako minimální hodnota, která dostačuje pro následný výpočet odhadu rozestupu, aniž by byla odchylka výsledné vzdálenosti větší, než 10 %. Závislost této odchylky na vybraném prahu je možné vidět na obrázcích 6.4. V levém grafu je zobrazen celý rozsah a na pravém je možno přesněji vyčíst odchylky pro vyšší práh **IoU**.



Obrázek 6.4: Závislost odhadu vzdálenosti na změně **IoU** (*Intersection over Union*). Na levém grafu je vyobrazen celý rozsah hodnot a na pravém je snížen rozsah odchylky pro přesnější výčet hodnot.

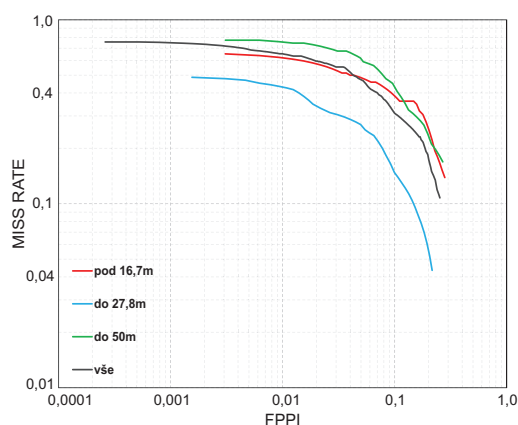
Celkové Vyhodnocení klasifikátoru vozidel je vyobrazeno ROC křivkami na grafu 6.5. Stejným způsobem probíhalo i vyhodnocení pro použitý klasifikátor registračních značek, které ukazuje graf 6.6.

Křivky pro jednotlivé rozsahy vzdáleností byly získány přidáním vypočteného odhadu rozestupu z obrazu do anotačního souboru. Následné vygenerování křivek proběhlo zvlášť pro definované rozmezí vzdáleností, které bylo vybráno na základě bezpečného dodržování rozestupu viz tabulka 6.4.

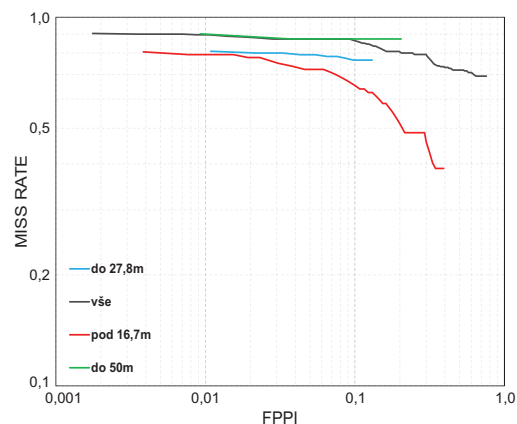
$$MissRate = \frac{\Sigma FN}{\Sigma P} \quad (6.2)$$

Jednotlivé ROC křivky představují závislost **Miss Rate** na **FPPI** (*False Positives Per Image*), viz vzorec 6.2. Hodnota **Miss Rate** zde udává podíl anotovaných objektů na snímcích, které se nepodařilo detekovat  $FN$ , ku celkovému počtu všech anotovaných objektů  $P$ .

Naopak **FPPI** je počet falešných detekcí ku všem snímkům. V této práci reprezentují anotované a detekované objekty vozidla a v druhé případě registrační značky.



Obrázek 6.5: ROC křivka - vyhodnocení detekce vozidel



Obrázek 6.6: ROC křivka - vyhodnocení detekce registračních značek

Na grafech jsou vyneseny křivky ukazující přesnost a spolehlivost detektoru. Čím více se křivka blíží vertikálně k nule, tím menší je celková hodnota **Miss Rate**, tedy tím větší je poměr True positive. Na druhou stranu, čím více směřuje křivka doprava, tím je větší **False Positive**, tudíž dochází k detekování obrazových oblastí, které nezachycují správný objekt. Rozsah hodnot obou os je zde omezen z důvodu lepší reprezentace výsledků, hodnoty obou os obvykle nabývají i větších hodnot. Taktéž bylo zvoleno logaritmické měřítko na ose x, aby byly zřejmé rozdíly jednotlivých křivek.

Při pohledu na oba grafy lze tedy vyčíst, že detektor vozidel, vytvořený v rámci této bakalářské práce, má mnohonásobně vyšší úspěšnost detekce, nežli detektor pro rozpoznávání registračních značek, který byl převzat z demo příkladů OpenCV knihovny<sup>1</sup>.

Toto chování lze zdůvodnit průměrnou kvalitou snímků, které byly získány z vytvořených video záznamů. Tudíž registrační značky nebyl detektor schopný tak dobře rozeznat. Tuto skutečnost je možné vidět na grafu 6.6, kde je viditelné zlepšení detekce reg. značek, při přiblížení vozidla ke kameře. Při porovnání úspěšností jednotlivých křivek, pro různé rozsahy vzdáleností, je znatelný trend lepší úspěšnosti se snižující se vzdálenosti detekované registrační značky.

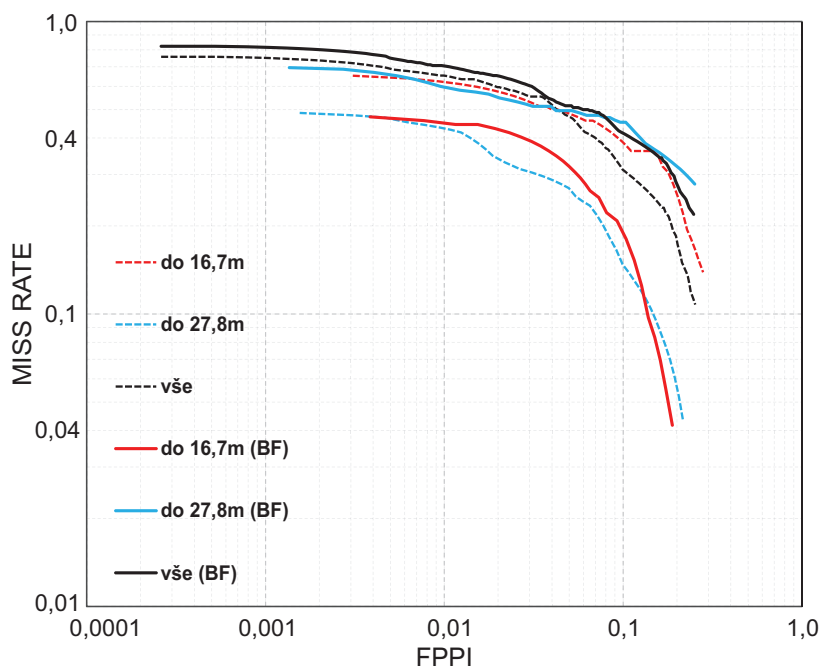
Natrénovaný detektor vozidel vykazuje ve výsledcích dobrou úspěšnost. Celkový Miss Rate detektoru se blíží hodnotě 0, viz graf 6.5. Oproti detektoru RZ však vykazuje poněkud odlišný trend závislosti vzdálenosti na úspěšnosti detekce. Detekce registračních značek se při menším rozestupu vozidel zlepšuje, avšak detektor vozidel dosahuje při středních vzdálenostech mezi 16,7 m až 27,8 m.

Na první pohled neočekávané chování nastalo při detekci vozidel na velmi krátké vzdálenosti (do 16,7 m). Při ručním procházení výsledků detekce, kde je vozidlo takto blízko kameře, bylo zjištěno, že detekce vozidla sice proběhla, avšak detekovaná oblast vozidla byla příliš malá oproti anotaci, tudíž parametr **IoU** byl nižší než vybraný práh, proto byla detekce ignorována.

Při hledání řešení pro tuto skutečnost bylo experimentálně použito silnějšího filtrování pomocí **bilaterálního filtru** pro odstranění digitálního šumu a vyhlazení obrazu. Díky

<sup>1</sup><https://github.com/opencv/opencv/blob/master/data/haarcascades/>

tomu bylo docíleno znatelně spolehlivější detekce, avšak pouze na velmi krátké vzdálenosti. Na rozestup větší než 20 m byla tato funkce pro vyhlazení obrazu spíše kontraproduktivní. Porovnání výsledků bez použití a s použitím **bilaterálního filtru** je možné vidět na následujícím grafu 6.7. Čárkované křivky představují výsledky bez použití filtru a plné křivky ukazují výsledky s použitím filtru.



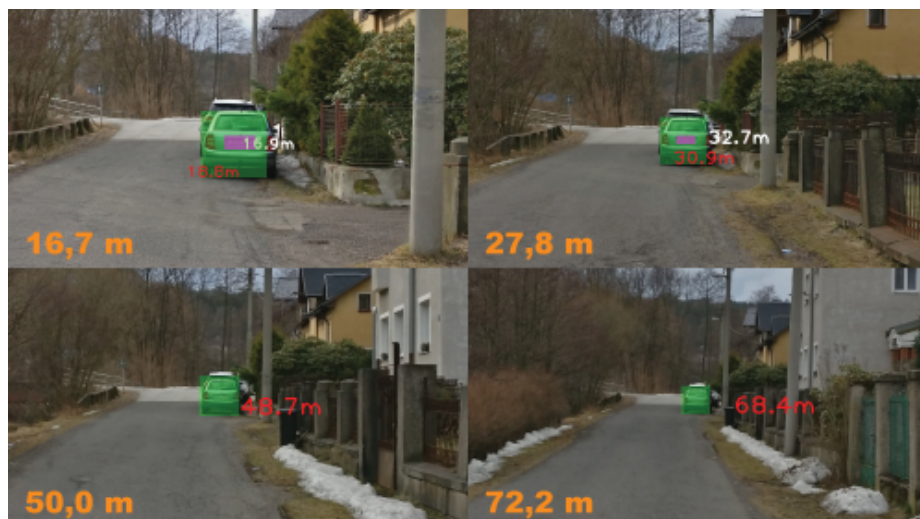
Obrázek 6.7: ROC křivka - porovnání detekce vozidel bez použití a s použitím bilaterálního filtru pro vyhlazení obrazu (označeno BF)

## 6.4 Vyhodnocení odhadu vzdálenosti

Pro vyhodnocení odhadu vzdálenosti pomocí mobilní aplikace bylo vytvořeno 9 sad statických fotografií vozidel a natočeno 25 minut videa z reálného provozu.

Referenční fotografie byly nafoceny čtyřmi různými mobilními telefony, abychom dokázali ověřit výpočet vzdálenosti vozidel, se zohledněním parametrů kamery a snímače mobilních telefonů. Každá z devíti sad snímků obsahuje fotografie automobilu, zachyceného ze čtyř předem definovaných vzdáleností udaných v tabulce 6.4. Tyto vzdálenosti byly při vytváření naměřeny 50 m pásmem. Pro každé vozidlo tedy vzniklo celkem šestnáct fotografií, pořízených čtyřmi telefony se záměrně odlišnými parametry. Ukázka z datové množiny je na obrázku 6.8.





Obrázek 6.8: Příklad vyhodnocení na sadě referenčních fotek (Červený odhad je z detekovaného vozidla, bílý odhad je z detekované registrační značky a oranžovou barvou je znázorněna opravdová vzdálenost)

Rychlost v [km/h]	Vzdálenost za 2 s [m]
30	16,7
50	27,8
90	50,0
130	72,2

Tabulka 6.4: Vzdálenosti ujeté během 2 sekund v běžných rychlostech v provozu

Vzdálenost ujetá za dvě sekundy byla zvolena na základě vyhlášky o bezpečného vzdálenosti<sup>2</sup>.

Výsledky vyhodnocení odhadu vzdálenosti z obrazu byly získány pomocí algoritmu popsaného ve vzorci 4.1. Následující tabulka výsledků 6.5 ukazuje, že odhad rozestupu z detekce registrační značky je v průměru 2× přesnější než odhad z detekce automobilu. Vyhodnocení detekce ale ukázalo, že detekce registračních značek není tak úspěšná, jako detekce vozidel, viz grafy 6.5 a 6.6.

typ detekce	průměrná odchylka [%]
vozidla	19,80
registrační značky	9,80

Tabulka 6.5: Vyhodnocení odhadu vzdálenosti bez použití regresní funkce

Z vyhodnocení bylo též zjištěno, že u mobilních telefonů s vyšším rozlišením se úspěšnost klasifikace automobilů snižovala i při zmenšení snímku algoritmem na rozlišení maximálně 1280 × 720. Pro řešení tohoto problému bylo použito mírného filtrování obrazu pomocí

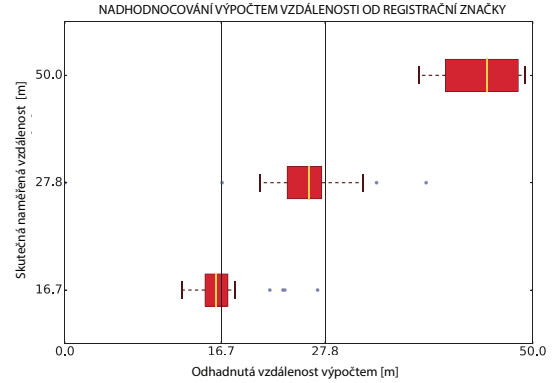
<sup>2</sup>[http://www.cspsd.cz/storage/files/vzdalenost\\_mezi\\_vozidly.pdf](http://www.cspsd.cz/storage/files/vzdalenost_mezi_vozidly.pdf)

bilaterálního filtru, který sice zjemní snímek, ale jen v jasově podobných oblastech, tím pádem nepřijdeme v obraze o ostrost hran a přechodů, ale jen se zbavíme moc ostrého a zašumělého obrazu. Tím bylo možné vylepšit klasifikaci vozidel a registračních značek.

První vyhodnocení vzdálenosti probíhalo následujícím způsobem. Pro každý snímek vyhodnocovacího datasetu statických fotografií byl vypočítán odhad vzdálenosti (viz výpočet 4.1) a jeho výsledek se přiřadil do množiny výsledků pro danou vzdálenost, ve které bylo vozidlo na daném snímku zachyceno. Následně na základě takto vytvořených množin výsledků byly vytvořeny box diagramy, které jsou na grafu 6.9. Na stejném principu probíhalo i vyhodnocení detekce vzdálenosti z detekce registrační značky vozidla, viz graf 6.10.



Obrázek 6.9: Graf ukazující nadhodnocování výsledků vzdáleností s použitím detekce vozidel



Obrázek 6.10: Graf ukazující podhodnocování výsledků vzdáleností s použitím detekce registračních značek

Na grafech jsou vertikálními čarami vyznačeny vzdálenosti, kterým se mají výsledky vyhodnocení blížit. Jak je ale možné vidět na pravém grafu, medián výsledků, reprezentovaný žlutou linií v boxech, je ve všech případech větší, než reálná vzdálenost. To znamená, že vzorec pro výpočet vzdálenosti vozidel v obraze nadhodnocuje výslednou vzdálenost. Podobný případ lze vidět i na levém grafu, kde je vidět opačné chování. Výsledná vzdálenost vypočítaná z detekcí registračních značek je naopak podhodnocována.

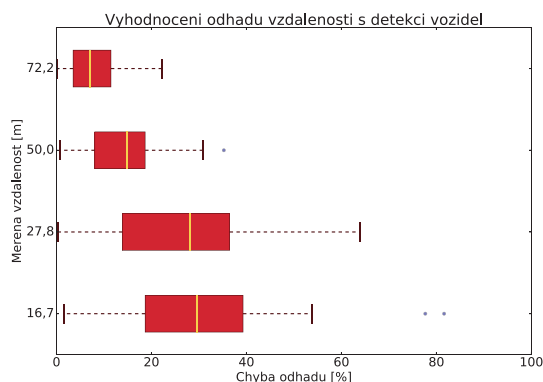
Výstupní grafy vyhodnocení na statických fotografiích ukázaly, že klasifikátor detekuje v průměru menší část vozidla a tedy nadhodnocuje vzdálenost. Naopak v případě detekce registračních značek bylo zjištěno z grafů opačné chování, kdy detektor klasifikoval i část okolí registrační značky. Obě tato zjištění vedla k návrhu řešení tohoto problému a vylepšení výpočtu vzdálenosti.

Řešením pro problém nadhodnocování a podhodnocování je použití vhodné regresní funkce pro výsledek výpočtu vzdálenosti, která by pomohla snížit nadhodnocení, či podhodnocení.

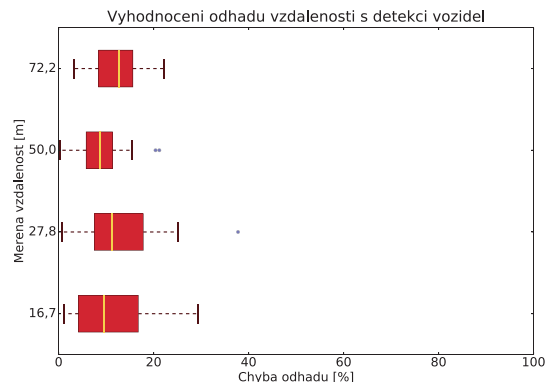
Jako nejlepší regresní funkce, která by minimalizovala chybu a zanesla by se do výpočtu rozestupu aplikace, byla zvolena lineární regrese. Při experimentování bylo zjištěno, že funkce  $y = k \cdot x + q$  s vypočítáním koeficientů  $k$  a  $q$  na základě výsledků poloviny vyhodnocovací množiny, sice přinese zlepšení v detekci automobilů, avšak výrazně zhorší odhad vzdálenosti s detekcí registračních značek. Po dalších experimentech nejlépe vyšla funkce  $y = k \cdot x$  s rozdílnými koeficienty  $k$  pro odhad s detekcí vozidel a s detekcí registračních značek. Důvodem pro volbu odlišných koeficientů je opačné chování, tedy nadhodnocování v případě odhadu z detekcí vozidel a podhodnocování s detekcí registračních značek.

Abychom mohli vyhodnotit výsledky výpočtu vzdálenosti s použitím regresní funkce, bylo nutné rozdělit množinu statických fotografií na dvě části. Na první se hledal vhodný koeficient, který by zlepšil výpočet a na druhé se následně regresní funkce vyhodnotila.

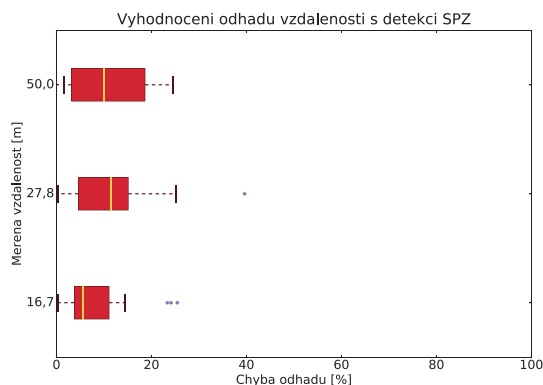
Porovnání vyhodnocení průměrné odchylky vzdálenosti, před a po zanesení regresní funkce do algoritmu výpočtu vzdálenosti, je vyobrazeno na následujících čtyřech grafech 6.11, 6.12, 6.13 a 6.14.



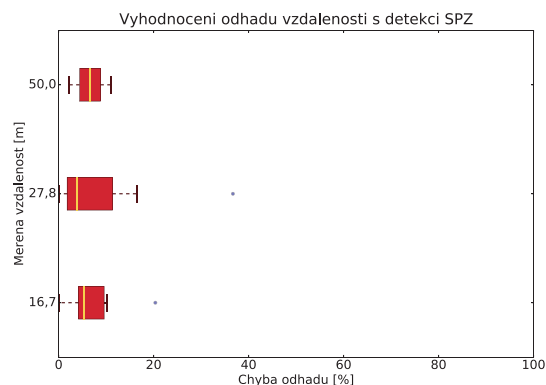
Obrázek 6.11: Graf vyhodnocení odhadu vzdálenosti při detekci vozidel (před použitím regresní funkce)



Obrázek 6.12: Graf vyhodnocení odhadu vzdálenosti při detekci vozidel (s použitím regresní funkce)



Obrázek 6.13: Graf vyhodnocení odhadu vzdálenosti při detekci registračních značek (před použitím regresní funkce)



Obrázek 6.14: Graf vyhodnocení odhadu vzdálenosti při detekci registračních značek (s použitím regresní funkce)

Z grafů je vidět, že použití regresních funkcí velmi pozitivně ovlivnilo chybovost výpočtu. Pro lepší představu snížení chyby výpočtu, byla vytvořena následující tabulka 6.6, ve které jsou porovnány průměrné odchylky odhadu s použitím regresní funkce a bez ní.

	průměrná odchylka [%]	
	bez regresní funkce	s regresní funkcí
vozidla	19,96	11,32
registrační značky	9,00	7,50

Tabulka 6.6: Vyhodnocení odhadu vzdálenosti na polovině vyhodnocovací sady před a po použití regresní funkce

## 6.5 Zhodnocení výsledků

Hlavním cílem této práce byl odhad rozestupu mezi vozidly z obrazu. Proto bylo navrženo řešení, které vypočítává vzdálenost na základě šířky detekovaných oblastí reprezentujících vozidlo či registrační značku. Vyhodnocení odhalilo problémy s nadhodnocováním výsledků při výpočtu z šířky vozidla a naopak podhodnocováním vzdálenosti při výpočtu z šířky registrační značky. Po použití regresních funkcí se však podařilo tento problém podstatně redukovat a snížit tak průměrnou odchylku téměř dvojnásobně. I tak je zde prostor pro zlepšení odhadu vzdálenosti. Výrazný vliv na přesnost odhadu má například ještě kvalita detekce oblasti vozidel a registračních značek. Tím je myšlena odchylka velikosti detekovaného bounding boxu oproti reálné oblasti vozidla. To jsem se snažil vyřešit použitím odhadu vzdálenosti z velikosti registrační značky, který je mnohem přesnější a kompenzuje tedy tento problém. Zlepšení by ale ještě mohlo přinést přetrénování kaskádového klasifikátoru s větší sadou negativních vzorků a především její větší rozmanitosti. Pomocť by mohlo taky vytvoření přesnějších výřezů vozidel ze snímků, které reprezentují pozitivní vzorky. Tím by se mohlo docílit větší robustnosti a přesnosti detekované oblasti.

V rámci práce byl také vytvořen detektor vozidel, založený na kaskádového klasifikátoru. Tento klasifikátor byl trénován na datové sadě obsahující velké množství různých druhů vozidel i v odlišných světelných podmínkách. Z vyhodnocení vyplynulo, že vytvořený detektor je poměrně spolehlivý, především na detekci vozidel ve střední vzdálenosti, ale o něco slabších výsledků dosahuje na velké vzdálenosti, ale také na velmi krátké vzdálenosti, což mě překvapilo. Zlepšení této situace přineslo použití bilaterálního filtru, ale i přesto došlo jen k malému zlepšení. Tento problém by dle mého názoru mohlo zlepšit rozšíření trénovací sady o větší a ostřejší vzorky vozidel, jelikož vzorky z použitého datasetu **TME Motorway Dataset**, které tvoří podstatnou část trénovacího datasetu, měly malé rozlišení.

## Kapitola 7

# Závěr

Cílem mé bakalářské práce bylo vytvořit aplikaci pro odhad rozestupu vozidel z obrazu kamery mobilního telefonu. Tento cíl se mi podařilo naplnit. Aplikace byla vytvořena pro systém android. Odhad rozestupu byl navrhnout a implementován a jeho výsledky byly popsány v kapitole o vyhodnocení. Přesnost odhadu je relativně úspěšná, odchylka odhadnuté vzdálenosti dosahuje průměru 7,50 % při výpočtu z šířky detekované registrační značky a 11,32 % při odhadu z šířky vozidla. Pro účely měření rozestupu byl tedy navrhnout výpočet vzdálenosti vozidla z obrazu, zohledňující parametry kamery, šířku detekovaného automobilu v obraze a fyzickou šířku vozidla, zastoupenou průměrnou šířkou vozidla. Stejný výpočet byl použit pro odhad na základě velikosti registrační značky, s tím rozdílem, že se do výpočtu zahrnuly rozměry registrační značky místo rozměrů vozidla.

Pro detekci vozidel byl vytvořen kaskádový klasifikátor natrénovaný na pozitivních a negativních datových sadách, které byly vytvořeny v rámci této práce. Pro detekci registračních značek byl využit již vytvořený kaskádový klasifikátor z volně dostupných demo příkladů OpenCV knihovny.

Cílem vytvořené mobilní aplikace bylo také ukládání záznamů z jízdy. Tento cíl byl také splněn, jeho řešení bylo popsáno v kapitole implementace.

Výsledná aplikace byla navíc ještě doplněna o experimentální řešení sledování rozestupu vozidel a zvukovou signalizaci při nebezpečném snižování rozestupu.

Jak bylo popsáno v poslední podkapitole o zhodnocení výsledků vyhodnocení, pokračování práce bych chtěl směřovat ke zmenšení odchylky velikosti detekovaných vozidel, s čím úzce souvisí následný odhad rozestupu a také ke zlepšení detekce na krátké vzdálenosti. Podstatné zlepšení odhadu vzdálenosti přineslo již přidání detekce registračních značek. Ještě lepším řešením by ale mohlo být vytvoření trénovací datové sady s robustnější negativní sadou vzorků. Pro řešení druhého problému, s detekcí na krátké vzdálenosti, jsem navrhl doplnění trénovací množiny o snímky vozidel z větší blízkosti, respektive s lepším rozlišením, jelikož současná pozitivní množina obsahuje takových vzorků málo.

S touto bakalářskou prací jsem se zúčastnil Studentské konference inovací, technologií a vědy v IT Excel@FIT 2017. Kde má práce získala dvě ocenění za výjimečnou práci.

# Literatura

- [1] Ahonen, T.; Hadid, A.; Pietikainen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 28, č. 12, Dec 2006: s. 2037–2041, ISSN 0162-8828, doi:10.1109/TPAMI.2006.244.
- [2] Caraffi, C.; Vojir, T.; Trefny, J.; aj.: A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera. In *ITS Conference*, Sep. 2012, s. 975–982.
- [3] Cika, P.; Zukal, M.; Sebela, M.: Vehicle license plate detection and recognition using symbol analysis. In *2011 34th International Conference on Telecommunications and Signal Processing (TSP)*, Aug 2011, s. 589–592, doi:10.1109/TSP.2011.6043661.
- [4] Deng, J.; Dong, W.; Socher, R.; aj.: ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [5] Duda, R. O.; Hart, P. E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM*, ročník 15, č. 1, Leden 1972: s. 11–15, ISSN 0001-0782, doi:10.1145/361237.361242.
- [6] Hartley, R.; Zisserman, A.: *Multiple View Geometry in computer vision*. The address: Cambridge University Press, druhé vydání, 2003, ISBN 9780521540513.
- [7] Janeček, M.: Soustava kamer jako stereoskopický senzor pro měření vzdálenosti v reálném čase. 2014.
- [8] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, editace F. Pereira; C. J. C. Burges; L. Bottou; K. Q. Weinberger, Curran Associates, Inc., 2012, s. 1097–1105.
- [9] Manaf A. Mahammed, F. A. K. ., Amara I. Melhum: Object Distance Measurement by Stereo VISION. *International Journal of Science and Applied Information Technology (IJSAIT)*, ročník 2, č. 2, March 2013: s. 05–08, ISSN 2278-3083.
- [10] PŘINOSIL, M., J.; KROLIKOWSKI: Využití detektoru Viola-Jones pro lokalizaci obličeje a očí v barevných obrazech. *Elektrorevue - Internetový časopis (www.elektrorevue.cz)*, , č. 31, 2008: str. 16, ISSN 1213-1539.
- [11] Wang, Y.-Q.: An analysis of the Viola-Jones face detection algorithm. *Image Processing On Line*, ročník 4, 2014: s. 128–148.

- [12] Žára, J.; Beneš, B.; Sochor, J.; aj.: *Moderní počítačová grafika*. Computer Press, 2004, ISBN 9788025104545.